Educational Materials

CMU/SEI-91-EM-5

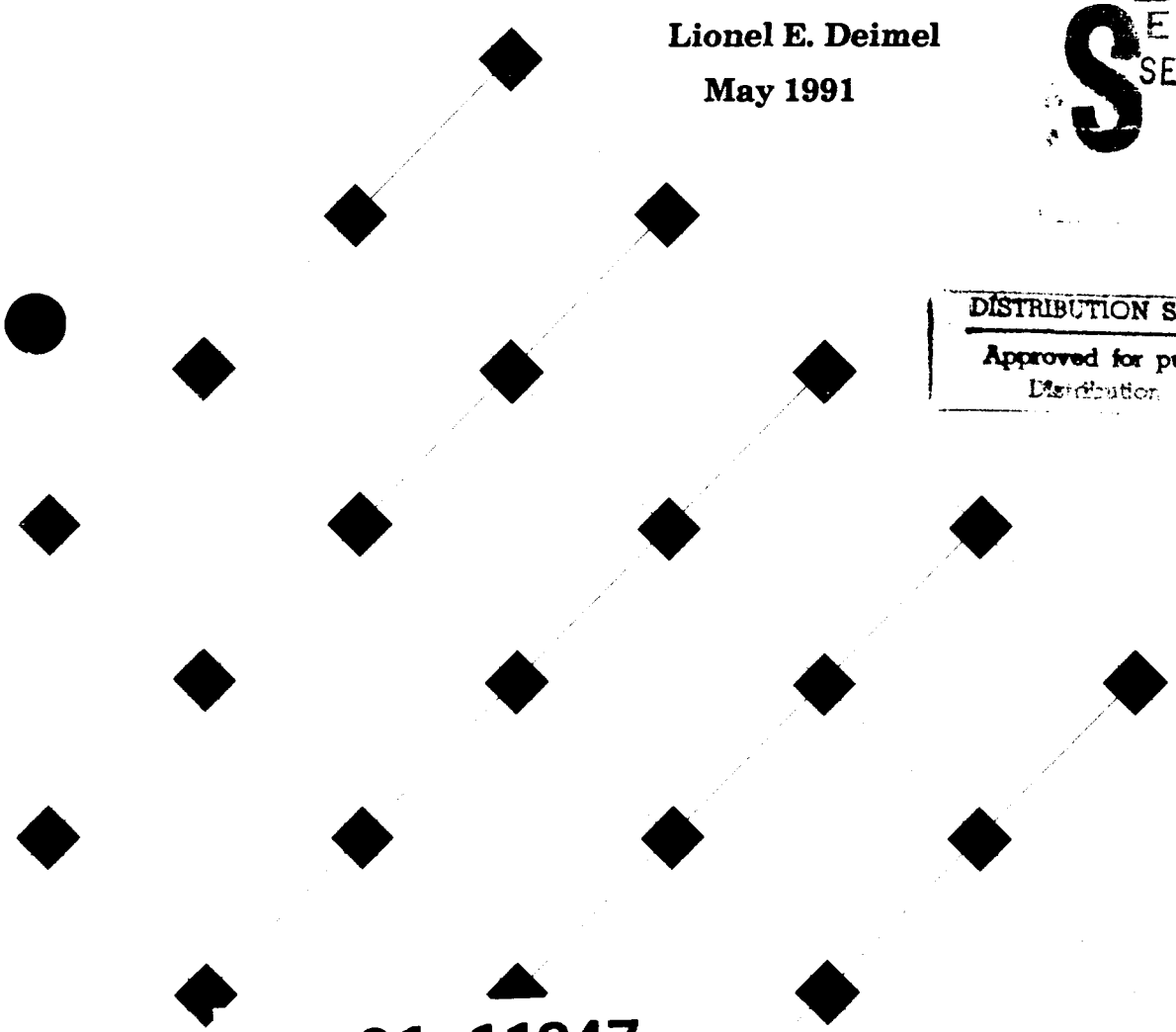
Carnegie-Mellon University
Software Engineering Institute

Scenes of Software Inspections

Video Dramatizations for the Classroom

Lionel E. Deimel

May 1991



The following statement of assurance is more than a statement required to comply with the federal law. This is a sincere statement by the university to assure that all people are included in the diversity which makes Carnegie Mellon an exciting place. Carnegie Mellon wishes to include people without regard to race, color, national origin, sex, handicap, religion, creed, ancestry, belief, age, veteran status or sexual orientation.

Carnegie Mellon University does not discriminate and Carnegie Mellon University is required not to discriminate in admissions and employment on the basis of race, color, national origin, sex or handicap in violation of Title VI of the Civil Rights Act of 1964, Title IX of the Educational Amendments of 1972 and Section 504 of the Rehabilitation Act of 1973 or other federal, state, or local laws or executive orders. In addition, Carnegie Mellon does not discriminate in admissions and employment on the basis of religion, creed, ancestry, belief, age, veteran status or sexual orientation in violation of any federal, state, or local laws or executive orders. Inquiries concerning application of this policy should be directed to the Provost, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, telephone (412) 268-6684 or the Assistant Provost for Enrollment, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, telephone (412) 268-2056.

Educational Materials

CMU/SEI-91-EM-5
May 1991

Scenes of Software Inspections

Video Dramatizations for the Classroom

Lionel E. Deimel

Software Engineering Curriculum Project

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

This document was prepared for the

SEI Joint Program Office
ESD/AVS
Hanscom AFB, MA 01731

The ideas and findings in this document should not be construed as an official DoD position. It is published in the interest of scientific and technical information exchange.

Review and Approval

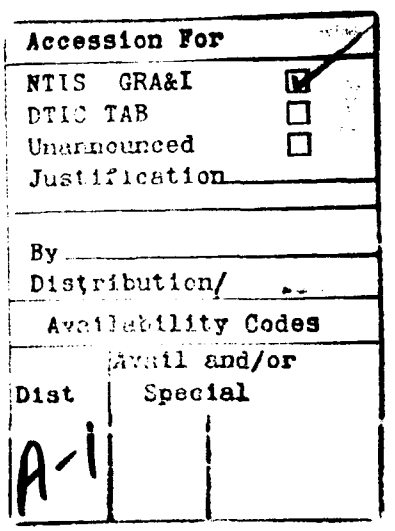
This document has been reviewed and is approved for publication.

FOR THE COMMANDER

Charles J. Ryan, Major, USAF
SEI Joint Program Office

The Software Engineering Institute is sponsored by the U.S. Department of Defense.
This report was funded by the Department of Defense.

Table of Contents



Preface

In 1988, under the leadership of Scott Stevens, the Advanced Learning Technologies (ALT) Project at the Software Engineering Institute (SEI) began to design and produce an interactive learning system using Digital Video Interactive (DVI) technology. The project staff chose software code inspection as the topic to be taught using this prototype system. To illustrate aspects of code inspections, a number of brief scenes were performed by professional actors, videotaped, and incorporated into the product. These vignettes are useful educational materials, even in the absence of a compact disc player and the special computer hardware required for the full DVI learning system. We have therefore transferred them to a standard VHS videotape and written this report, which explains what is on the videotape and how it might be used in teaching about software inspections.

I would like to thank the ALT staff for its video production and its cooperation in making this material available to a wider educational audience. Scott Stevens and Michael Christel provided ALT production documents and acted as reviewers. Judy Chiswell, now with Recursive, offered useful insight into the vignettes themselves, assisted with the bibliography, and reviewed the final report.

This material was also reviewed by Maribeth Carpenter, Gary Ford, Priscilla Fowler, Linda Levine, and Linda Pesante, all of the SEI, and by Charles Weber, SEI resident affiliate from the Federal Sector Division of IBM. I am indebteded to each of these reviewers for comments that led to improvements of earlier drafts.

Finally, I would like to express my appreciation to Kurt Haverstock who edited a number of videotapes for me over a period of several weeks, usually on short notice, but always with good cheer; to John Antonucci, who edited the final tape; and to Anita Harnish, of Ugly Dog Studios, who helped with the credits.

— L.E.D.

Scenes of Software Inspections
Video Dramatizations for the Classroom

Abstract: This report describes the videotape *Scenes of Software Inspections*, which contains brief dramatizations that demonstrate appropriate and inappropriate conduct of software inspections. The tape also includes scenes that show other kinds of group interactions. Any of these scenes can be incorporated into lectures, self-study materials, or other educational delivery mechanisms, to illustrate how to perform inspections, an important software engineering technique.

1. Introduction

Since they were first described by Michael Fagan in 1976, software inspections have received widespread recognition as an effective technique to identify and eliminate defects from software and software-related artifacts. In an inspection, a small peer group of software developers systematically "reads" a document, identifying and classifying defects as they are encountered. Participants play specific roles in this process, which is designed not only to improve product quality, but also to make the software production process more visible and more controllable.

Software inspections resemble other types of software technical reviews, such as walkthroughs, in that they involve the examination and discussion of software work products by a group of software developers. It is important to emphasize, however, that inspections are distinctive, and the inspection process is formal and relatively inflexible. Inspections are characterized by: explicit entry and exit criteria; individual preparation by inspectors; defined roles of moderator, reader, producer, and recorder; training for moderators; use of a checklist; limitation of discussion to identification and classification of defects; a requirement that successful completion of rework is necessary to complete the inspection; and formal data collection, reporting, and analysis. The literature suggests that the impressive effectiveness of inspections at improving product quality is diminished if elements of the process are omitted or modified.

This report describes scenes depicted on a companion videotape. Seven of these scenes dramatize properly and improperly conducted inspections of code written in a high-level language; four illustrate other types of group activities. Educators can use these vignettes to illustrate lectures, or incorporate them into self-study packages, or study them to improve their own understanding of the inspection process and other group processes. In the next two sections, a description of each scene is accompanied by remarks and suggestions to help instructors use the videotape effectively.

The scenes are described in the order in which they appear on the videotape. Section 2 describes scenes that illustrate the software inspection process itself. The scenes can be used to show what an inspection is and what can go wrong in an inspection. Section 3 describes additional scenes of groups of people interacting with one another. These scenes are useful for introducing the idea that we function as members of many kinds of groups, each with its own goals and rules of behavior. Section 4 offers guidance in the selection of scenes and use of the tape.

Readers are assumed to have some familiarity with software inspections. The brief annotated bibliography at the end of this report can be used to find additional information. This bibliography is not meant to be a complete guide to the inspection literature, but it does contain some of the more important references, as well as references concerned with teaching about inspections and using inspections in teaching.[1] A few references dealing with group processes are also included.[2]

2. Inspection Scenes

The first seven scenes portray events in the inspection process. Although code inspections are illustrated, the activities shown are typical of software inspections, irrespective of the particular product being reviewed. Scene 1 illustrates a well-run inspection; Scenes 2-6 present various problems that can occur in an inspection. All six scenes show software developers inspecting the same two pieces of code written in a high-level language. Scene 7 shows the moderator of the inspection discussing problems with her project manager.

That participants play well-defined roles is an important feature of software inspections. Understanding these roles and recognizing who is acting in which capacity are essential to appreciating fully the action shown. On the videotape, the inspectors (moving in a counter-clockwise direction) are

- Marie, the *moderator*
- Rick, the *reader*
- Pete, the *producer*
- Mike, the *recorder*

The *moderator* is the chief planner and meeting manager. The moderator has the responsibility of ensuring that the inspection process is faithfully executed. In particular, the moderator must maintain the focus of the review on finding defects.

---

[1]The original paper on inspections is [Fagan76]. Readers looking for a quick introduction to inspections should read [Ackerman89] or [Russell91]. In addition, [IEEE89] is invaluable for clarifying what a software inspection is and is not. Both [Gilb88] and [Humphrey89] offer strong arguments for inspections and practical advice about conducting them.

[2]Of the works cited, [Brilhart89] is the most general and the most up-to-date.

The *reader* leads the inspectors through the work product under review by reading or paraphrasing it section by section, or even line by line. The reader sets the pace of the inspection, although the moderator has the responsibility of making sure that this pace is appropriate.

The *producer* is the author of the work product under review and, as such, has the responsibility of meeting the entry criteria for inspection. The producer performs any rework on the product deemed necessary by the inspection team. During the inspection, the producer contributes his or her special insight as author of the work product.

The *recorder* is responsible for documenting the defects identified in the inspection.

These roles are usually—but not invariably—performed by different people. However, the author of the work product being inspected cannot play any of the above roles except that of producer.

All participants also act as *inspectors,* and there may be additional reviewers present acting only in this capacity. Inspectors identify and describe defects in the work product. Each inspector must review the product in advance and be familiar with inspection procedures. Inspectors may be chosen to represent particular viewpoints, for example, that of maintenance personnel.

Scene 1: Beginning of a Successful Code Inspection

LENGTH: 2 min., 38 sec.

This scene shows the beginning of a smoothly running inspection of two modules, **Tank_Check** and **Operator_Feedback**. Marie, the moderator, states the purpose of the inspection and checks that everyone has received all necessary materials. She asks for and records everyone's preparation time. The reader, Rick, begins paraphrasing the header of **Tank_Check**. Mike observes that certain parameters are redundant. The group agrees with this analysis; and Mike, the recorder, notes the defect as "unnecessary code." Rick continues reading as the scene fades.

Scene 1 illustrates the minimum preliminaries required at an inspection review. Notice that the moderator performs no introductions, does not explain who is playing which role, does not describe the nature of an inspection, and does not review the inspection checklist. All these activities are unnecessary if the inspectors know each other and are well-versed in inspection techniques.

If one were to pick a single scene from the videotape to illustrate a software inspection, this would be the scene. It shows how to begin an inspection and conveys some sense of the basic procedure used to identify and record defects. This scene can provide students with a more tangible model of how an inspection is conducted than can a purely verbal description.

Since the moderator, reader, producer, and recorder are not explicitly identified in this scene, an instructor can play the scene for a class that is just learning about inspections and can ask who is playing which role. The instructor should also ask what cues students used to identify the participants. This exercise helps clarify the nature of the roles in students' minds. In this scene, by the way, there is an early, subtle cue that Pete is the author of the modules being inspected—his declared preparation time is much less than that of the other inspectors.

Notice that all participants in the inspection are ready for the work they are doing and are very businesslike in their demeanor. Everyone is relaxed; the producer is not defensive about his code; and the other inspectors are constructive in their remarks. The group is assembled to enhance the quality of the product, not to show off or engage in petty squabbles. As should always be the case with software inspections, no managers are present.

This scene and Scene 3 illustrate the technique of reading or paraphrasing code. Students should practice this technique, which requires more skill than might first be apparent. The inspection literature emphasizes that the rate of reading is critical—it should be neither too fast nor too slow. A reader, not the producer, performs this role. This arrangement frees the producer to concentrate on bringing his or her special perspective to the group, and avoids the producer's representing intentions as the actual product.

It can be useful to ask students how events might proceed differently if a design, test plan, or piece of documentation were being inspected. For each type of work product, questions such as the following can be asked:

- How does one prepare for the inspection?
- How does the reader read or paraphrase?
- What kinds of defects can be found by inspection?
- What kinds of defects are unlikely to be found by inspection?
- What should be on the inspection checklist?

Thinking about and discussing these issues can be important for student learning, but nothing has quite the impact of having students perform practice inspections, in which they must confront the issues operationally. Practice inspections should be followed by postmortem analysis of what took place in the review.

Scene 2: Moderator Dominates Inspection

LENGTH: 2 min., 4 sec.

The moderator of an inspection is charged with keeping the inspection team focused on its task and presiding fairly over the review. In this scene, however, the moderator abuses her role. Marie talks too much and constantly interrupts others, behaviors likely to sabotage the effectiveness of the inspection.

This and the next four scenes show various ways an inspection can go astray. Each scene can be used either to introduce or to illustrate a particular kind of problem. For example, the instructor can show a scene and then talk about the problem it portrays. A more exciting approach is to show the scene and ask students what *they* see, thereby beginning a dialogue aimed at helping them to appreciate and understand the problem. If the class is well prepared to discuss inspections, having either read or heard about them, the scenes can be used as part of a game. In this case, the instructor shows the scene, and students vie to see who will be the first person to name the problem encountered by the inspection team. Of course, each scene can also be used to illustrate graphically a problem the instructor has just discussed.

It is particularly useful to discuss with students what they see going on in Scene 2. Some students may view Marie as simply over-enthusiastic. In fact, she is acting like an intellectual bully who is intent upon having her views heard and who is not particularly interested in what others have to say. In one case (whether to put code to raise the tank pressure in a separate procedure), the group assents to her suggestion, seemingly to humor her. In another case (the discussion of a variable name), her need to dominate the discussion inhibits rather than facilitates the process of articulating defects. In addition, Marie violates a cardinal rule of inspections by engaging in problem solving—she spends time trying to find the "right" identifier. (See also Scene 5, in which the producer engages in problem solving.)

What we see here illustrates a risk inherent in Marie's roles of moderator and inspector. If she expresses her views as an inspector but fails to treat her contribution and those of other participants with equity, she can convert an essentially democratic process into an authoritarian one. An important goal of moderator training is teaching future moderators how to avoid this pitfall.

The results of Marie's poor behavior as moderator may not appear serious. Studying the faces of the other inspectors, however, will suggest there is trouble ahead. Rick, Pete, and Mike are clearly frustrated by Marie's behavior. At one point they seem to be looking at one another and thinking, "Here she goes again!" The likely outcome of this inspection is that the other inspectors will let Marie talk about whatever she wants and will themselves participate as little as possible. That way, they can get out of an unpleasant and unsatisfying situation as quickly as possible. Needless to say, the quality of the inspection will suffer.

Once students appreciate the inappropriateness of Marie's behavior, there is an opportunity to bring them to a deeper understanding of the dynamics of the inspection. Try asking questions such as:

- Judging by their actions and expressions, what do you think is going through the minds of Rick, Pete, and Mike?
- How would you feel if you were participating in this inspection? What would you want to do or say?

- As an inspector, what could you do to rescue the meeting from an overbearing moderator? How do you think the moderator would react to your intervention? How would the moderator feel?
- Why do you think Marie is acting as she is?
- Have you ever behaved like Marie? How easily could you fall into acting like Marie if you were the moderator of an inspection?

Scene 3: Reader Is Too Fast

LENGTH: 1 min., 2 sec.

The reader paces an inspection. It is his or her responsibility to divide the work being inspected into manageable pieces the inspection team can deal with. The reader can err either by focusing too much on detail or by lumping together so much material that important details are glossed over. In this scene, Rick tries to cover too many lines of code at once. When he says that the code on lines 20-35 "checks for a valid temperature," both Pete and Mike object that they have questions about a number of lines within the loop in question. The moderator suggests that Rick slow down so that the code can be examined in greater detail, and the inspection proceeds.

In this scene, it is clear that Rick is trying to cover too much material at once. In practice, this mistake is not always immediately obvious. It is the moderator's responsibility to ensure that the reading rate is appropriate for the type of material being inspected, even when other inspectors do not notice the reader's error. For code in a high-level language, the ideal rate is about 100-150 documented lines of code per hour.

Notice that Marie is gentle in correcting Rick, and Rick is good-natured about accepting the criticism.

It is worth noting how, at the very end of this scene, Mike consults Pete because Pete "know[s] this part of the program better than any of us." The producer is being asked to contribute his special insight as author.

Scene 4: Producer Is Under Attack

LENGTH: 1 min., 2 sec.

In inspections, as in other forms of software technical reviews, it is important to review the *product*, not the *producer*. In this scene, however, Mike implies that Pete does not know how to write proper comments for his code. Pete takes offense, but his doing so fails to discourage Mike from making more pointed comments. The moderator finally steps in and reminds everyone that the objective is to review code, "not each other," and the inspection proceeds.

It may not be immediately obvious to students that Mike's remarks are inappropriate, or even that they are personal. Apparently, there *is* a problem with line 23, as indicated by the fact that Rick, the reader, has trouble describing exactly what the code does. Mike jumps in to say, "Again, without good supporting code comments, we've got a mess on our hands." (Apparently the question of the adequacy of comments has come up previously.) In saying this, Mike has not merely identified a defect; he has also subtly expressed his own emotional reaction in the guise of an "analysis" of the situation. Finding another defect in the code is hardly an adequate explanation for the strength of his reaction—the real message is, "Pete messed up again!" Pete, who probably was put on the defensive by Mike's earlier remarks, rightly construes this remark as a personal attack. Realizing that matters are getting out of control, Marie gently tries to calm everyone. But Mike does not quit, ending his next remark with, "How are we supposed to follow this mess?" Pete counterattacks with the remark that the code is "self-documenting." The implication: Mike is not smart enough to figure it out. Mike is not very responsive to this: "Yeah, we see, we know. This is the worst mess we've ..." Mike is interrupted by Marie, who more assertively plays her role as moderator and reminds everyone that personal assaults are inappropriate.

This scene helps students appreciate the complexity of discourse within a group. An instructor may need to show the scene several times and ask leading questions before students begin to articulate the *real* messages being sent by the participants. A good way of analyzing the interaction is to step through the scene, isolating each line spoken. After every statement, one can ask:

- What is being said?
- Why did the speaker say that?
- What is the speaker thinking?
- What is the speaker feeling?
- What is the message behind the statement?

At the end of the scene, the instructor can then ask what actions could have been taken by whom to defuse the situation earlier. Some students will find this a very difficult exercise; avoiding personal remarks is more difficult than many people imagine.

Scene 5: Producer Begins Problem Solving

LENGTH: 0 min., 58 sec.

The task of participants in a software inspection is to identify defects in the inspected work product, not to decide what to do about them. Not only can a group be quite inefficient at designing fixes to identified problems, but time spent on problem solving decreases the time available for locating additional defects. This scene shows how easy it is for an inspection team to fall into problem solving rather than defect identification. Mike suggests that a temperature check in the code is implemented in the wrong place. Pete, who wrote the code, immediately recognizes this as a problem and begins scan-

ning his listing to decide what to do about it. While doing so, he is thinking aloud. It is not clear whether he is talking to himself or to Mike, but he clearly is not talking to the group. Mike makes a specific suggestion, and for a moment the two of them work on the problem together. Finally, Marie interrupts the conversation to remind Mike and Pete that the team is supposed to be finding defects, not removing them.

The temptation to solve a problem once it is found is almost irresistible. Computer professionals, who see themselves as problem solvers, find it difficult to believe that engaging in problem solving can be other than a good thing. It does seems a shame to suppress Pete's enthusiasm, but it is important for Marie to intervene. One of the reasons—apart from the time lost for defect identification by every member of the inspection team—can be found in Rick's expression of boredom. His time is being wasted as Mike and Pete carry on their private conversation, and his attentiveness to the inspection is likely to decrease if the moderator is too permissive about digressions from the inspection agenda.

Notice that the moderator is tactful in bringing the team back to its task. Marie acknowledges Pete's competence and suggests taking the discussion off-line. Pete and Mike are gracious in accepting Marie's reminder.

Not being able to share constructive insights can indeed be very frustrating to inspectors. Recognizing this, some organizations set aside time after the formal inspection, during which interested members of the inspection team can discuss solutions to the problems found. Ultimately, it is the responsibility of the producer to choose how defects are to be corrected, subject to review by the moderator or to re-inspection.

The problem encountered by the inspection team is named very late in this particular scene, when Marie reminds the group that their objective is merely to *find* defects. For this reason, the instructor may want to stop the tape where Marie begins speaking, and ask the students what has gone wrong.

Scene 6: Recorder Is Too Slow

LENGTH: 0 min., 47 sec.

For the inspection process to run efficiently, all inspectors must be adequately prepared. In this scene, Mike is having difficulty with his role as recorder. He seems not to have understood a problem identified by the group and is therefore having difficulty writing it down. Despite an explanation from Pete, Mike still looks puzzled. The moderator suggests that Mike should be better prepared and should pay closer attention in the future.

The scene does not show clearly why Mike is having the difficulty he is, and it is worth discussing whether Marie is as tactful as she might be in this situation. The moderator has the responsibility to terminate an inspection if the participants are not prepared,

and Marie may be displaying some understandable anxiety. The recorder's role is admittedly very demanding. If the recorder has not carefully studied the work product under review, he or she may not be able to follow the discussion and understand the defects identified. Also, if the recorder's mind wanders, he or she may fail to get all the details right in the description written on the inspection report. (See also Scene 7.)

Scene 7: Inspection Is Analyzed

LENGTH: 1 min., 17 sec.

In this scene, the project manager has come to Marie's office to discuss why certain errors were not removed from the product, despite its having undergone code inspections. Marie explains that the problems were in fact found, but they were not recorded in sufficient detail to facilitate proper rework. She agrees that more care should be taken to ensure that adequate information is captured. In turn, the manager agrees to increase training for recorders and establish better procedures to avoid such problems.

Instructors may want to ask students how credible they find this scene. To some students, particularly those with job experience, the action may seem more idealistic than realistic. Managers are not always dispassionate about the failures of their technical staffs, nor so willing to commit to additional training as a mechanism for improving performance. A well-run inspection process, however, necessarily includes monitoring of its effectiveness and a willingness to make changes in the process if warranted. Events such as those in Scene 7 should be more common in real life than they are.

It is also worth asking students if Marie and her manager are being completely honest with one another. Marie's explanation that the fault lies with the recorder may be something of a face-saving measure. Not only is the *moderator* responsible for ensuring that rework is done properly, but the entire inspection team had the opportunity to catch the vagueness in the problem descriptions when findings were reviewed at the close of the inspection meeting. It is, in fact, counter to the spirit of inspections to place blame on a single team member; the inspection team as a whole is responsible for the quality of the product. The project manager is either naive about the inspection process or knows Marie well enough to know that a gentle suggestion that preserves her pride is sufficient to get her to correct the process she is responsible for controlling.

3. Group Process Scenes

For software inspections to be successful, the members of the inspection team must function effectively as group members. Our educational system does an excellent job of teaching us how to compete with one another, but it is less successful in teaching us how to cooperate and work together toward a common goal. As a result, instruction in the conduct of software inspections can often benefit from discussion of group process

and group communication. Scenes 8-11, which show people functioning in various group contexts, can be used to facilitate such discussion. The scenes illustrate the three major types of groups:

- Social groups
- Family groups
- Task-oriented groups (shown functioning well and shown in conflict)

These scenes were written with a particular educational objective in mind, namely, to bring students to the realization that we are members of various groups that function differently and have different purposes. Because of this, reflection on our own experiences can be a starting point for understanding the functioning of groups generally. Moreover, Scenes 8-11 are useful for placing the very structured interactions of the software inspection within the broad context of group dynamics.

A software inspection is carried out by a task-oriented group. That group is not formed to socialize, provide support to one another, or educate. Its purpose, instead, is to find defects in software and related artifacts. The group is a very specialized kind of task-oriented group, one in which—unlike the group shown in Scene 10—the roles of participants are precisely defined. Of course, it is important to recognize that even task-oriented groups are populated by people with human needs and failings. The emotional dimension cannot be ignored, and it sometimes even dominates interactions, as in Scene 11.

For each of the scenes described below, students can be asked to discuss questions such as:

- Why are these people part of a group?
- What are the purposes of this group?
- What are the "rules" for behavior in this group?
- How does a group of this type differ from groups of other types?
- How do the participants feel about the action portrayed?
- What do you think happens after the scene we are shown?

Scene 8: Social Group

LENGTH: 1 min., 3 sec.

Social groups are often organized in an informal way and have primarily social goals, such as maintaining affiliation. This scene illustrates such a group. Two couples are packing a car for a tailgate party and football game. The conversation revolves around whether Nicole, who is preparing the food for the party, will make the group late or will bring so much food that they will have to use a larger car.

The couples in this scene might be said to be performing a task, but what they are really doing is getting together to socialize and have fun. The task in which we see them engaged is simply a means to that end. There is a degree of tension and conflict in this scene; but because this is a social group and not a group formed to "accomplish" something, it would be considered inappropriate for one of the group members to react too strongly. Were this a task-oriented group, however, it would be quite in order for a member of the group to intervene assertively to get everyone to the football game faster. Were it a family group, some teaching or discussion about responsibility might be in order.

Three of the four characters in Scene 8 appear again in later scenes. Nicole is in Scene 9, and her husband Bill is in Scenes 9-11. Their friend Dave reappears in Scenes 10-11. The same characters are shown repeatedly to emphasize that each of us participates in a variety of groups in everyday life.

Scene 9: Family Group

LENGTH: 0 min., 50 sec.

Family groups provide the environment in which we learn love and trust, experience acceptance, and develop our sense of self-worth. Although both social and family groups can provide support and a sense of affiliation, the strong and enduring ties of family can best provide a safe and supportive haven from life's difficulties. This scene shows Nicole and Bill talking to their son about a problem he is having in a college class.

As in Scene 8, the principals in this scene are ostensibly engaged in a task, namely, washing windows. However, the important activity here is neither window washing nor socializing. Instead, the parents are providing loving support for their son, and teaching him—or more likely reminding him—how to avoid discouragement and how to stand up for himself.

In addition to the general questions suggested earlier, possible questions for student discussion are:

- When would a family group engage in a more task-oriented function?
- From your own experience, what are the key differences between your participation in family and social groups?
- What influence does family group experience have on task-oriented work experience?

Scene 10: Task-Oriented Group

LENGTH: 1 min., 1 sec.

Task-oriented groups in industry have specific goals or tasks to accomplish. This scene shows Bill, a group leader, conducting a meeting to decide whether to automate a manufacturing process.

This scene demonstrates group leadership and functional group interaction. The discussion is calm, rational, and to the point. Bill is careful to allow each person a chance to speak and express concerns. Everyone is consulted and supported in offering his or her expertise.

Compare this scene to Scene 11, in which the same group engages in destructive and counterproductive behavior.

Specific discussion questions include:

- How does leadership influence group activity?
- What types of behavior support and encourage group participation?
- When do group activities begin to lose focus?
- What positive experiences have you had in task-oriented groups? Describe the effects.

Scene 11: Task-Oriented Group in Conflict

LENGTH: 0 min., 40 sec.

Here we see the same group as in Scene 10. It is now about 15 minutes later, and the group is no longer functioning well. In contrast to the previous calm discussion, the meeting has degenerated into a shouting match involving Larry, manager of quality control; Dave, the production manager; and Holly, a design engineer.

Several aspects of the action in this scene are worth noting. Perhaps most important is the fact that Bill, the leader of the meeting, is now ineffective at controlling it. Although he attempts to interrupt the argument and regain the floor, everyone ignores him. (Were Bill the moderator of an inspection and behaved this way, we would consider his performance unsatisfactory.) Bill is the only person who is really addressing the issue at hand. Larry, Dave, and Holly are more interested in berating one another about past performance than in addressing today's concerns. (Holly implies that Larry makes poor estimates; Dave accuses Holly of either stating unrealistically low estimates or being unable to control costs; Holly complains that Larry and Dave keep changing their minds; and Larry, in exasperation, attacks Holly's competence as a design engineer.) This company seems to have serious problems that need to be addressed if it is to function smoothly, but this free-for-all is not a good mechanism for addressing them.

Much of the communication that takes place in this scene is nonverbal, and students can have some fun analyzing it. Pay special attention to the tone of voice and gestures of the participants. Note, for example, how Dave expresses his incredulity by throwing his head to the side, and how Holly reinforces her accusations with her pencil.

Through careful circumscription of behavior, the software inspection process is engineered to avoid the kind of conflict dramatized in this final scene. Instructors may wish to discuss questions such as the following:

- Why might the moderator of an inspection have more "authority" to control arguments than Bill does in this scene?
- How have the participants broadened the agenda and thereby worked themselves into this argument?
- Where do the loyalties of the participants lie? Are they operating as a team?
- How can Bill regain control of the meeting?
- How could the "rules" for such meetings be changed to avoid unproductive conflict in the future?

4. Scene Selection

Few users of the videotape are likely to employ all 11 scenes in the order provided. Although that order is logical, it is not based primarily on pedagogical concerns. This section, therefore, offers some suggestions to educators and trainers about selecting scenes.

The tape is structured to make it easy to move from scene to scene. So that instructors retain the option of not giving students cues as to what each scene is about, scenes are innocuously titled "Scene 1," "Scene 2," etc. These 15-second lead-ins can be used for scene identification when scanning the tape. Relying on this feature to move between vignettes at either end of the tape, however, can take several minutes. Users can save time by noting in advance the counter or timer index at the beginning of each scene to be shown and then using the high-speed forward and reverse functions. (Because of differences between VCRs, this should be done on the VCR that is going to be used to show the tape.) Scenes may also be copied to a new tape in a different order.[3]

The sets of scenes described in Section 2 and Section 3 can be used individually or together. Some instructors may choose to show only the scenes described in the earlier section, to facilitate teaching about inspections. Because software engineering requires technical people to work in groups to accomplish a variety of tasks, however, others may use the group process scenes as a way of introducing the idea of working together

[3]See permission statement on the inside back cover of this document.

on a project, independent of performing inspections. For example, Scenes 10-11—or perhaps Scenes 8-11—can be used at the beginning of a one-semester software engineering course to help students understand what it means to work on a team project.

Instructors wanting to teach both inspections and group process can treat either topic first or can interleave them. For students with some experience as members of task-oriented work groups, a good strategy might be to provide an introduction to group process, followed by extended discussion of software inspections as a specific instance of the task-oriented group. For students without work experience, it may be more appropriate to introduce inspections first. A more general treatment of groups can follow, drawing examples both from students' personal experiences and from their understanding of inspections.

A few additional words are in order about the scenes illustrating the inspection process. Scene 1 illustrates a well-run inspection review. This can be a helpful scene to show in contexts where the educational objective is knowledge of what an inspection is, rather than achievement of a deep understanding of its dynamics. Scenes 2-6 explore the pathology of inspections, showing some—but by no means all—of what can go wrong. If students are to gain a real understanding of the inspection method and if time permits, all these scenes should be shown and discussed. Scene 7 touches on what happens when things go wrong, but it also makes the point that the inspection process—like other software engineering processes—should be constantly monitored, analyzed, and optimized for maximum effectiveness.

Seeing dramatizations of inspections is surely more compelling than reading or hearing about them, but it is not a substitute for actually experiencing the inspection process. Although *Scenes of Software Inspections* can be an effective introductory aid, instructors who want their students to understand the power of inspections should provide opportunities for them to participate in real or practice inspections as well.

Annotated Bibliography

Ackerman83

Ackerman, A. Frank, Priscilla J. Fowler, and Robert G. Ebenau. "Software Inspections and the Industrial Production of Software." In *Software Validation: Inspection-Testing-Verification-Alternatives*, Hans-Ludwig Hausen, ed. Amsterdam: North-Holland, 1984, 13-40.

Abstract: *Software inspections were first defined by M.E. Fagan in 1976. Since that time they have been used within IBM and other organizations. This paper provides a description of software inspections as they are being utilized within Bell Laboratories and the technology transfer program that is being used for their effective implementation. It also describes the placement of software inspections within the overall development process, and discusses their use in conjunction with other verification and validation techniques.*

A rather lengthy description of the introduction and conduct of software inspections at Bell Laboratories. The paper is most interesting for its discussion of the effort required to introduce inspections. The volume in which this paper appears is the proceedings of the Symposium on Software Validation, held September 25-30, 1983, in Darmstadt, Germany.

Ackerman89

Ackerman, A. Frank, Lynne S. Buchwald, and Frank H. Lewski. "Software Inspections: An Effective Verification Process." *IEEE Software 6*, 3 (May 1989), 31-36.

An excellent brief introduction to what inspections are and how they are performed.

Brilhart89

Brilhart, John K., and Gloria J. Galanes. *Effective Group Discussion*, 6th Ed. Dubuque, Iowa: Wm. C. Brown, 1989.

This popular textbook examines the dynamics of small groups, both for the student interested in what is known about group process and for the practitioner seeking practical guidance. Readers need no special background in group dynamics.

Collofello87

Collofello, James S. "Teaching Technical Reviews in a One-Semester Software Engineering Course." *ACM SIGCSE Bulletin 19*, 1 (February 1987), 222-227.

Abstract: *Software technical reviews are essential to the development and maintenance of high quality software. These review processes are complex group activities for which there exist an abundance of basic concepts evolved over years of practical experience. In a typical one-semester software engineering course very little of this information is adequately conveyed to students. Texts supporting this course are also very weak in this area. This paper provides a practical approach for teaching about software technical reviews in a one-semester software engineering course. The contents for two to three lectures on this topic are described as well as suggested exercises and an approach for integrating technical reviews with the usual team project. An extensive annotated bibliography is also provided to assist instructors and students.*

Practical advice about introducing software technical reviews, including inspections, into a one-semester software engineering project course.

Collofello88
Collofello, James S. *The Software Technical Review Process.* Curriculum Module SEI-CM-3-1.5, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pa., June 1988.

Capsule Description: *This module consists of a comprehensive examination of the technical review process in the software development and maintenance life cycle. Formal review methodologies are analyzed in detail from the perspective of the review participants, project management and software quality assurance. Sample review agendas are also presented for common types of reviews. The objective of the module is to provide the student with the information necessary to plan and execute highly efficient and cost effective technical reviews.*

This SEI curriculum module discusses software inspections in the context of all varieties of software technical reviews. This particular curriculum module has a heavy emphasis on teaching advice.

Cross88
Cross, John A., ed. *Support Materials for The Software Technical Review Process.* Support Materials SEI-SM-3-1.0, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pa., April 1988.

This package, a supplement to [Collofello88], contains teaching materials designed to support the teaching and classroom use of software technical reviews, and particularly inspections. Included are teaching suggestions, student handouts, forms, and checklists.

Daniels86
Daniels, William R. *Group Power: A Manager's Guide to Using Meetings.* San Diego: University Associates, 1986.

A brief, atheoretical guide to improving meeting effectiveness. Although the book does not address inspections directly, it reinforces the idea that groups can be powerful problem solvers and provides exercises to encourage sensitivity and competence in interactions within task-oriented groups.

Fagan76
Fagan, Michael E. "Design and Code Inspections to Reduce Errors in Program Development." *IBM Systems J. 15,* 3 (1976), 182-211.

This is Fagan's original paper describing inspections at IBM. Several authors have called this paper a "classic." The growing literature on inspections makes it less necessary reading than formerly, however.

Fagan86
Fagan, Michael E. "Advances in Software Inspections." *IEEE Software SE-12,* 7 (July 1986), 744-751.

Fagan's most recent update on software inspections.

Freedman82
Freedman, Daniel P., and Gerald M. Weinberg. *Handbook of Walkthroughs, Inspections, and Technical Reviews,* 3rd Ed. New York: Little, Brown, 1982.

In a question-and-answer format, the authors discuss inspections and related review processes, such as walkthroughs. This handbook, [Collofello88], and [IEEE89] are particularly helpful to anyone interested in comparing various review methods. The book also contains a detailed discussion by Gary Shelly and Thomas Cashman on the use of walkthroughs in the classroom; this may also be useful to instructors teaching software inspections.

Gilb88

Gilb, Tom. *Principles of Software Engineering Management.* Wokingham, England: Addison-Wesley, 1988.

Chapter 12 of this book, "The Inspection Process: Early Quality and Process Control," is devoted to inspections. Gilb, a strong advocate of inspections, provides a thorough and practical introduction that is very accessible to students.

Hare65

Hare, Alexander Paul, Edgar F. Borgatta, and Robert F. Bales, eds. *Small Groups: Studies in Social Interaction*, Rev. Ed. New York: Knopf, 1965.

This classic collection of papers on small groups provides the reader with historical and theoretical background on the subject and suggests the diversity of issues and research.

Hollocker90

Hollocker, Charles P. *Software Reviews and Audits Handbook.* New York: John Wiley, 1990.

A very practical guide to software audits and reviews, including inspections, written by the chairman of the effort that produced [IEEE89]. About half the book consists of process and product checklists, sample forms, sample reports, and the like. Several pages of references are also included.

Humphrey89

Humphrey, Watts S. *Managing the Software Process.* Reading, Mass.: Addison-Wesley, 1989.

Humphrey provides not only a chapter on inspections (Chapter 10) but also an appendix, "Conducting Software Inspections." He is very thorough and explicit. Instructors can make good use of the inspection forms reproduced in the appendix.

IEEE89

IEEE. *IEEE Standard for Software Reviews and Audits.* ANSI/IEEE Std 1028-1988, corrected edition. New York: IEEE, June 30, 1989.

This standard discusses inspections and other review processes within a common framework. This approach is helpful for comparing and contrasting different methods. The section on software inspections provides a succinct description of the essential elements of the inspection process.

Russell91

Russell, Glen W. "Experience with Inspection in Ultralarge-Scale Developments." *IEEE Software 8*, 1 (January 1991), 25-31.

This paper is an excellent introduction to inspections, how to perform them, and how to introduce them into an organization. The author discusses his experience with inspections at Bell-Northern Research.

Yourdon89
Yourdon, Edward. *Structured Walkthroughs*, 4th Ed. Englewood Cliffs, N.J.: Yourdon Press, 1989.

Although this book is not about inspections, Yourdon's discussion of psychological factors in peer reviews applies equally well to inspections as to walkthroughs. The discussion is from the viewpoint of transactional analysis.

Videotape Order Form

Copies of the VHS videotape *Scenes of Software Inspections* are available from the Software Engineering Institute. The cost to U.S. Government agencies and to colleges or universities within the United States is **$25.00** for each tape ordered. The cost to all others is **$50.00**. Orders should be addressed to:

Education Program
Software Engineering Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213-3890

Checks should be made payable to **Carnegie Mellon University/SEI**. Please indicate if additional copies of this report are needed also.

Please send _____ copies of VHS tape @ ☐ **$25.00** ☐ **$50.00** each.

☐ Please include report with each copy.

Amt. Enclosed _____

Send to:

Name _____

Address _____

_____

_____

_____

Telephone _____

E-mail _____

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS None
2a. SECURITY CLASSIFICATION AUTHORITY N/A	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for Public Release Distribution Unlimited
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A	
4. PERFORMING ORGANIZATION REPORT NUMBER(S CMU/SEI-91-EM-5	5. MONITORING ORGANIZATION REPORT NUMBER(S) Educational Materials
6a. NAME OF PERFORMING ORGANIZATION Software Engineering Institute 6b. OFFICE SYMBOL (if applicable) SEI	7a. NAME OF MONITORING ORGANIZATION SEI Joint Program Office
6c. ADDRESS (City, State and ZIP Code) Carnegie Mellon University Pittsburgh PA 15213	7b. ADDRESS (City, State and ZIP Code) ESD/AVS Hanscom Air Force Base, MA 01731
8a. NAME OFFUNDING/SPONSORING ORGANIZATION SEI Joint Program Office 8b. OFFICE SYMBOL (if applicable) ESD/AVS	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER F1962890C0003
8c. ADDRESS (City, State and ZIP Code) Carnegie Mellon University Pittsburgh PA 15213	10. SOURCE OF FUNDING NOS. PROGRAM ELEMENT NO 63752F; PROJECT NO. N/A; TASK NO N/A; WORK UNIT NO. N/A
11. TITLE (Include Security Classification) Scenes of Software Inspections: Video Dramatizations for the Classroom	
12. PERSONAL AUTHOR(S) Lionel E. Deimel	
13a. TYPE OF REPORT Final 13b. TIME COVERED FROM TO	14. DATE OF REPORT (Yr., Mo., Day) May 1991 15. PAGE COUNT 20
16. SUPPLEMENTARY NOTATION	

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse of necessary and identify by block number)
FIELD	GROUP	SUB. GR.	code inspection, software quality assurance, review technologies, software engineering education, software inspection

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

This report describes the video tape, *Scenes of Software Inspections*, which contains brief dramatizations that demonstrate appropriate and inappropriate conduct of software inspections. The tape also includes scenes that show other kinds of group interactions. Any of these scenes can be incorporated into lectures, self-study materials, or other educational delivery mechanisms, to illustrate how to perform inspections, an important software engineering technique.

(please turn over)

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED ☒ ME AS RPTDTIC US ☐ S	21. ABSTRACT SECURITY CLASSIFICATION Unclassified, Unlimited Distribution
22a. NAME OF RESPONSIBLE INDIVIDUAL John S. Herman, Capt, USAF	22b. TELEPHONE NUMBER (Include Area Code) (412) 268-7630 22c. OFFICE SYMBOL ESD/AVS (SEI

ABSTRACT —continued from page one, block 19

The Software Engineering Institute (SEI) is a federally funded research and development center, operated by Carnegie Mellon University under contract with the United States Department of Defense.

The SEI Software Engineering Curriculum Project is developing a wide range of materials to support software engineering education. A *curriculum module* (CM) identifies and outlines the content of a specific topic area, and is intended to be used by an instructor in designing a course. A *support materials* package (SM) contains materials related to a module that may be helpful in teaching a course. An *educational materials* package (EM) contains other materials not necessarily related to a curriculum module. Other publications include software engineering curriculum recommendations and course designs.

SEI educational materials are being made available to educators throughout the academic, industrial, and government communities. The use of these materials in a course does not in any way constitute an endorsement of the course by the SEI, by Carnegie Mellon University, or by the United States government.

Comments on SEI educational materials and requests for additional information should be addressed to the Software Engineering Curriculum Project, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213. Electronic mail can be sent to education@sei.cmu.edu on the Internet.

Curriculum Modules (* Support Materials available)

CM-1 [superseded by CM-19]
CM-2 Introduction to Software Design
CM-3 The Software Technical Review Process*
CM-4 Software Configuration Management*
CM-5 Information Protection
CM-6 Software Safety
CM-7 Assurance of Software Quality
CM-8 Formal Specification of Software*
CM-9 Unit Testing and Analysis
CM-10 Models of Software Evolution: Life Cycle and Process
CM-11 Software Specifications: A Framework
CM-12 Software Metrics
CM-13 Introduction to Software Verification and Validation
CM-14 Intellectual Property Protection for Software
CM-15 Software Development and Licensing Contracts
CM-16 Software Development Using VDM
CM-17 User Interface Development*
CM-18 [superseded by CM-23]
CM-19 Software Requirements
CM-20 Formal Verification of Programs
CM-21 Software Project Management
CM-22 Software Design Methods for Real-Time Systems*
CM-23 Technical Writing for Software Engineers
CM-24 Concepts of Concurrent Programming
CM-25 Language and System Support for Concurrent Programming*
CM-26 Understanding Program Dependencies

Educational Materials

EM-1 Software Maintenance Exercises for a Software Engineering Project Course
EM-2 APSE Interactive Monitor: An Artifact for Software Engineering Education
EM-3 Reading Computer Programs: Instructor's Guide and Exercises
EM-4 A Software Engineering Project Course with a Real Client (forthcoming)
EM-5 Scenes of Software Inspections: Video Dramatizations for the Classroom